Cryptography

이훈재

hjlee@dongseo.ac.kr

http://crypto.dongseo.ac.kr

http://kowon.dongseo.ac.kr/~hjlee

Overview

- History of Cryptography (and Steganography)
- Modern Encryption and Decryption Principles
- Symmetric Key (Conventional) Cryptography
- Cipher Block Modes
- Key Management for Conventional Cryptography
- Message Authentication
- Public Key Cryptography
- Digital Signatures
- Key Management for Public-Key Cryptography

History of Steganography and Cryptography

Steganography

- Being able to communicate secretly has always been considered an advantage
  - Secret messages were often not written down, but rather memorized by sworn messengers
- Or hidden
  - Demaratus, a Greek immigrant to Persia, reveals Persia's intention to attack Athens. Write the secret message on a tablet, and covers it with wax.
  - Histaiaeus encourages Aristagoras of Miletus to revolt against the Persian King. Writes message on shaved head of the messenger, and sends him after his hair grew
  - Chinese wrote on silk, turned into wax-covered ball that was swallowed by the messenger
- Steganography
  - Steganos = "covered" in Greek, Graphein = "to write"

Steganography (cont.)

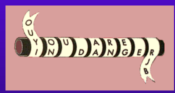
- Invisible Ink
  - Certain organic fluids are transparent when dried but the deposit can be charred and is then visible
  - A mixture of alum and vinegar may be used to write on hardboiled eggs, so that can only be read once shell is broken
- Embedded information
  - Germans used "microdots" - documents shrunk to the size of a dot, and embedded within innocent letters
  - Secret messages within music (Beatles)

Steganography (cont.)

- Steganography is also used to foil piracy in digital content
  - Watermarking copyright information into images, music
  - Programmers sometime embed "easter eggs"
- Steganography has been used by spies and children alike
  - Most recently, US argued that Bin Laden implanted instructions within taped interviews
- Steganography is weaker than cryptography because the information is revealed once the message is intercepted
- However, steganography can be used in conjunction with cryptography

Cryptography

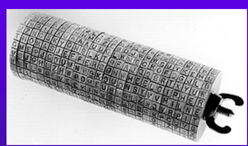
- In Cryptography, the *meaning* of the message is hidden, not its existence
  - Kryptos = “hidden” in Greek
- Historically, and also today, encryption involves
  - *transposition* of letters
    - Sparta’s scytale is first cryptographic device (5^{th} Century BC)
      - Message written on a leather strip, which is then unwound to scramble the message



  - *substitution*
    - Kama-Sutra suggests that women learn to encrypt their love messages by substituting pre-paired letters (4^{th} Century AD)
      - Cipher – replace letters
      - Code – replace words

Historical Cryptographic Exemplars

- Julius Caesar liked encrypting messages
  - Replaced Greek letters for Roman letters
- Caesar Shift Cipher
  - Each letter substituted by shifting *n* places
    - E X A M P L E
    - H A D P S O H
  - Only 25 such ciphers



- Substitution based on key phrase
  - Substitution key consists of phrase’s letters (uniquely) followed by rest of the alphabet
    - THIS IS ALICE AND BOB’S KEY
    - THISALCENDBOKY-FGJMPQRUVWXZ
  - 26! (roughly 10^{26}) monoalphabetic substitution ciphers

Historical Cryptographic Exemplars

- The Arabs broke monoalphabetic substitution using frequency analysis
  - In English (Beker&Piper)

a	8.2%	j	0.2	s	6.3
b	1.5	k	0.8	t	9.1
c	2.8	l	4.0	u	2.8
d	4.3	m	2.4	v	1.0
e	12.7	n	6.7	w	2.4
f	2.2	o	7.5	x	0.2
g	2.0	p	1.9	y	2.0
h	6.1	q	0.1	z	0.1
i	7.0	r	6.0		

  - Thus, letters ciphering e, t, and a are easily discovered
  - Subsequently can look for the rest of the letters and letter pairs

Historical Cryptographic Exemplars

- Homophonic substitution cipher can be used to foil frequency analysis
  - Keyed 2-digit substitution

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y/Z
T	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	00	01	02	03	04	05
H	43	44	45	46	47	48	49	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
E	71	72	73	74	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
K	90	91	92	93	94	95	96	97	98	99	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89

  - Reverse frequency

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	00	01	02	03	04	68	05
43	44	45	46	47	48	49	25	26			29	30	31	32	33		35	36	37	38		40		87	
71		73	74	50			53	54			57		59	60			63	64	65	66					
90			93	94			97	98			76		78	79			82	83	84						
72				51			56	58					61	34			39	86	42						
91				95			81	77					80	62			67	88	70						
92				52										85					89						
75				96															41						
				27															69						
				55																					
				99																					
				28																					

Historical Cryptographic Exemplars

- Vigenere's polyalphabetic cipher (16th century) generalizes Caesar's shift cipher
  - Can alternate between lines; or
  - Use keyword
- The Vigenere cipher is not amenable to simple frequency analysis

Vigenere Square

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	S
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Historical Cryptographic Exemplars

- Babbage broke Vigenere's Cipher (19th century)
  - Stage 1: Discover key length
    - Look for repeated sequences, and measure the distance between them
    - The key length is a factor of these distances
  - Stage 2: Identify the key itself
    - Compare distributions for each of the key letters with the standard distribution, to identify the shift

Historical Cryptographic Exemplars

- Coding
  - Louis XIV's Great Cipher (Rossignols) used one symbol (3-digit number) per syllable (held 200 years)
  - Mary Queen of Scots used a combination of cipher and coded words (nomenclator)
  - e.g,

assassinate	=	D	general	=	Σ	immediately	=	08
blackmail	=	P	king	=	Ω	today	=	73
capture	=	J	minister	=	ψ	tonight	=	28
protect	=	Z	prince	=	θ	tomorrow	=	43

Plain message = assassinate the king tonight
Encoded message = D–Ω–28

  - US Army used Navajo language as code in WWII

Transposition Ciphers

- Railfence: TRHCEEIETGSSMAIAEASS

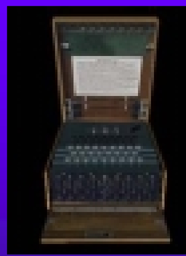
```
T 5  T                   R
H 3    H               C   E               E
E 1      I           E       T           G
K 4        S       S           M       A
E 2          I   A               E   S
Y 6            S                   S
```

- Redfence (by key): IETGIAESHCEESSMATRSS
- Columnar
  - IEEIRSHSMESCSTATGSEA

T	H	E	K	E	Y
5	3	1	4	2	6
T	H	I	S	I	S
A	S	E	C	R	E
T	M	E	S	S	A
G	E				

The German Enigma Machine (Scherbius)

- Electrical encryption machine, performing 8 substitutions
  - uses *n=3* rotating scramblers (26^n orientations)
  - scramblers can be configured in n! orders
  - pre-keyboard *k=6* swapped letter-pairs
- Encryption consists of
  - optional letter pair switch
  - compounded scrambling, with shifts
  - reflector swaps letter pairs
  - and backward scrambling
- Scrambles rotate in tandem
- Total of 10^{17} possible configurations
  - changed daily according to a codebook
  - each message has own orientation (message key)
  - later added 4th scrambler
- Used extensively by Germany in WW2
- Hitler used a more complex version of Enigma called Lorenz cipher

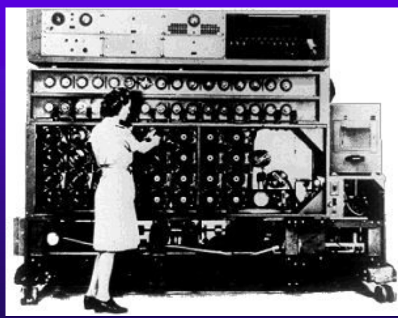


Poles Crack the Enigma

- Polish cryptanalysts obtained information about the encryption procedure from commercial Enigmas
- Obtained information on its usage
  - the Germans used a different *orientation* key for each message, encrypted twice in the message header (using the day key)
- Rejewski focused on the repetitions
  - Formalized relationships between 1st-4th ,2nd-5th, and 3rd-6th letters
    - ABCDEFGHIJKLMNOPQRSTUVWXYZ
    - FQHPLWOGBMVRXUYCZITNJEASDK
  - Built chains
    - (AFW), (BQZKVELRIB), (CHGOYDPC), (JMXSTNUJ)
  - Chains depend only on scrambler orientation, not pair swaps
    - Thus need to consider only 6 x 26^3 = 105456 configurations
  - Built a catalog of characteristic chains for all configurations

Poles Crack the Enigma

- Rejewski's algorithm to discover the day key
  - First, use catalog to identify the scrambler setting and orientation
  - Then, run the ciphertext through an Enigma and look at the text to identify swapped letter pairs
- Bombe machines were constructed to mechanize the search



British Crack Improved Enigma

- In 1939, Germans increased Enigma security
  - added 2 extra scramblers to choose – 10x arrangements
  - increased to 10 letter pair swaps
- British Cryptanalysts (Bletchley Park) took from the Polish
  - Recruited best Mathematicians (Turing) and large staff (7000)
  - Received Bombes from Polish
- Used human weaknesses provided hints and *cribs*
  - Trivial message keys (key sequences, names initials)
  - Artificial "intelligent" restrictions on scramblers arrangements and pair swaps restricted the search space
  - Standard message formats, e.g., weather
  - Some German codebooks were captured
- Turing constructed swap-independent chains similar to Rejewski
  - First British Bombe (*Victory*) delivered in 1940
  - Search still required significant human help
- The British ULTRA – broken German, Italian and Japanese communications were crucial to winning the war

Unbreakable Encryption

- One-time pads
  - Sender and receiver use a pre-arranged random stream of letters
  - Encryption=addition modulo 26
  - Every letter in the key used once

M	E	S	S	A	G	E
T	H	I	S	K	E	Y
F	L	A	K	K	K	C

- Perfectly secure encryption (Shannon)
  - Used by Soviet spies, and also for US-Soviet hotline
- Requires significant logistical effort and coordination
- Relies on randomness of key

Summary

- Encryption Algorithms and Keys
  - Substitution : bits, letters, words
  - Transposition
- Decryption Algorithms
  - Reversed process
  - Knowledge of the algorithm and the key
- Cryptanalysis
  - Identify algorithm
  - Obtain *as many* plaintext-ciphertext pairs
  - Use systematicity (patterns)
  - Use cribs

Modern Encryption and Cryptanalysis Principles

Main source: Network Security Essentials / Stallings

Modern Encryption Principles

- Encryption scheme has 5 ingredients
  - Plaintext, Encryption Algorithm, Key, Ciphertext, and Decryption Algorithm
  - Security depends on secrecy of the key, not algorithm

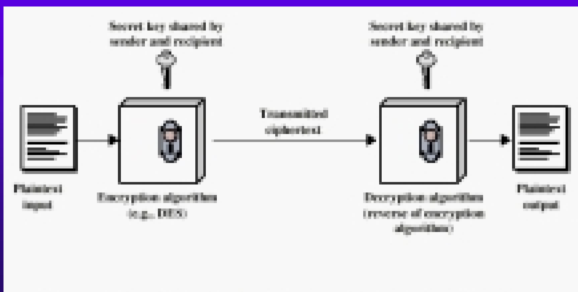


Figure 2.1 Simplified Model of Conventional Encryption

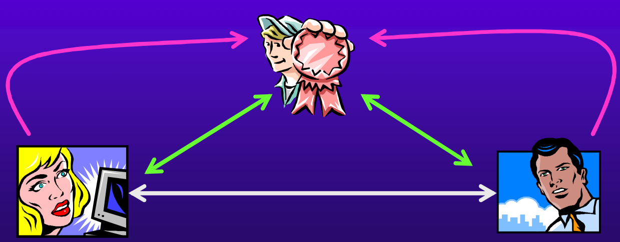
Notation

- M, or P will usually denote the plaintext message
- C will usually denote the ciphertext
- K will usually denote a key
- $E_k(M)=C$ is the encryption function
- $D_k(C)=M$ is the decryption function

- $D_k(E_k(M))=M$ represents the typical flow

Cryptographic Protocols

- Self enforcing protocols
- Arbitrated protocols
  - Trusted third party helps in real time
- Adjudicated protocols
  - Trusted third party, but only if needed and after the fact



Attacks Against Cryptographic Protocol

- Passive attacks (eavesdropping)
  - Cryptanalysis
  - Traffic analysis
- Active attacks
  - Impersonation
  - Interruption / denial
  - Modification of messages
  - Fabrication of new messages
  - Replay / Reflect messages

Cryptographic Algorithms

- Type of operations applies to plaintext
  - *Substitution* and *transposition*
- Type of key(s)
  - *Symmetric* : same key
  - *Asymmetric*, *Public-Key* : $D_{k2}(E_{k1}(M))=M$
- How plaintext is processed into ciphertext
  - How many and which operations
  - How the operations are combined
  - *Block ciphers*, *Stream ciphers*

Cryptanalysis (attacks against cryptographic algorithm)

- Ciphertext only
  - Uses only knowledge of algorithm and ciphertext
- Known plaintext
  - Also one or more plain-ciphertext pairs
  - Or, probable words: dictionary, known formats, etc.
- Chosen text
  - Chosen to reveal information about the key
  - Chosen plaintext and its ciphertext
    - Differential chosen plaintext
    - Adaptive chosen plaintext
  - Chosen ciphertext and its original plaintext
    - Mostly against public-keys

Computationally Secure Encryption

- Encryption scheme is *computationally secure* if
  - The cost of breaking the cipher exceeds the value of the encrypted information; or
  - The time required to break the cipher exceeds the useful lifetime of the information
- Most schemes that we will discuss are not unbreakable in principle, but are computationally secure
  - Usually rely on very large key-space, impregnable to brute force
- Moreover, the most advanced schemes rely on lack of knowledge of effective algorithms for certain hard problems, not on a proven inexistence of such algorithms
  - Usually factorization, discrete logarithms, or square roots mod p

Shannon's Theory of Secrecy

- Message *entropy* = minimum number of bits needed to express all possible messages
  - English entropy is 1.3 bits per letter
- Cryptanalysts try to modify the *a priori* probabilities of alternative messages until one emerges

- A cryptographic scheme is *perfectly secure* if knowledge of the ciphertext does not change the odds in favor of any of the possible plaintexts
- Shannon's Theory: the key must be at least as large as the message (entropy) and cannot be reused
  - Therefore, the secrecy of a cryptographic scheme depends on its entropy, i.e. the number of key bits, or the size of the key space
  - Only the one-time pad achieves perfect secrecy

Symmetric Key (Conventional) Cryptography

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Protocol

- Typical protocol
  - Alice and Bob agree on cryptosystem
  - Alice and Bob agree on a key
  - Alice encrypts her message with the key
  - Alice sends the message to Bob
  - Bob decrypts the messages using same key
- Variation
  - Alice selects a new key for each message and encrypts it using the agreed key
  - Alice sends the message key to Bob who decrypts it using the agreed key
  - Thereafter, Alice uses the message key to encrypt the actual message

Feistel Networks

- Most block encryption algorithms use this general structure, due to Horst Feistel (1973)
- Inputs: Plaintext (halved) , Key, Round function F
- Uses *n* rounds, in each
  - Inputs: L_i and R_i
  - $L_{i+1}=R_i$
  - $R_{i+1}=L_i \oplus F(R_i,K_i)$
  - F is a function that selects certain bits, duplicates some, and permutes them. K_i is derived from K
- Final ciphertext is combination of L_n and R_n
- At IBM, Feistel built *Lucifer*, the first such system

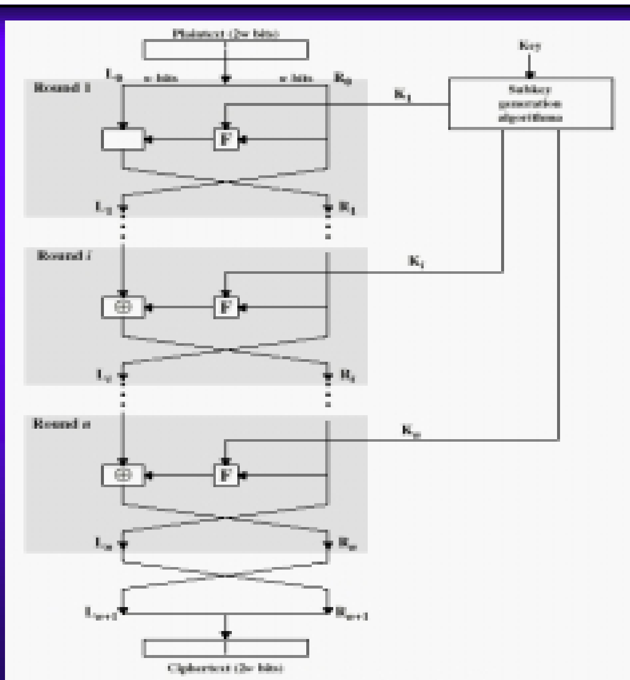


Figure 2.2 Classical Feistel Network

Notes on Feistel Cipher Structure

- Process is reversible
  - $R_{i-1}=L_i$
  - $L_{i-1}=R_i \oplus F(R_{i-1},K_{i-1})$
  - Same algorithm can be used but with keys reversed
- Security Considerations
  - Larger block size means fewer blocks and greater security
  - Larger key size means greater security
  - More rounds considered to offer better security (?)
  - Greater complexity of subkey generation may help security
  - Greater complexity of round function may increase security

Block Cipher Design Issues

- Easy to design a secure block cipher
  - By increasing the complexity of F (e.g., more complex S-boxes)
  - By iterating 1000 rounds
- Goals
  - Fast – few rounds, use simple operations
    - Low communication overheads
    - Low battery consumption in hand-helds
  - Easy to implement in hardware
    - Simple, ubiquitous operations
  - Efficient in memory usage
    - Can run on a smart card
  - Does not require too much secret material (keys, boxes)
    - Sometimes put on expensive tamper-proof memory

Data Encryption Standard (DES)

- Without a standard, software and hardware cannot interoperate, or at least it is very expensive
- In 1973, National Institute for Standards and Technology (NIST) issued RFP for Data Encryption Algorithm (DEA)
  - provide high level of security
  - completely specified and easy to understand
  - the security must reside in the ky
  - available to all users
  - adaptable to diverse applications
  - economically implementable in hardware
  - efficient to use
  - validated
  - exportable

Data Encryption Standard (DES)

- NIST (NBS) issued a Request For Proposal (RFC)
- IBM had only serious proposal
  - Patented and based on Lucifer (Feistel et al)
- NIST issued a Request For Comments (RFC)
  - Quite a few were concerned about NSA backdoor
  - NSA reduced the key size from 112 to 56 bits
    - Diffie and Helman presented a $20MM 1-day DES cracking machine
  - NSA had also changed the original S-boxes design
    - There were some claims of linearity in the new design
- DES was adopted in 1977

- In 1987, under NSA pressure, DES almost not recertified
- Until 1994, only hardware implementations of DES were permitted

Data Encryption Standard (DES)

- A Feistel block cipher structure
  - 64-bit blocks
  - 56-bit keys
  - 16 rounds
  - Adds initial and final permutation of the text (irrelevant to security)
  - Key shifted circularly for next round, and 48 bits are selected for K_i

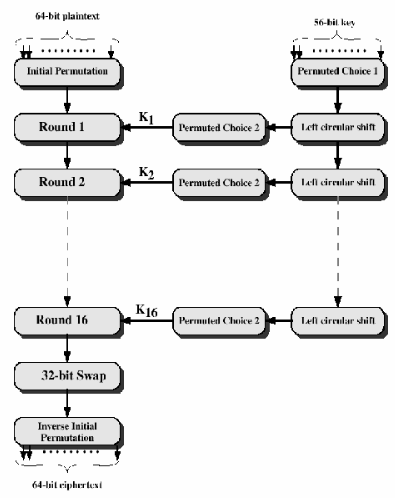


Figure 2.3 General Depiction of DES Encryption Algorithm

One Round of DES

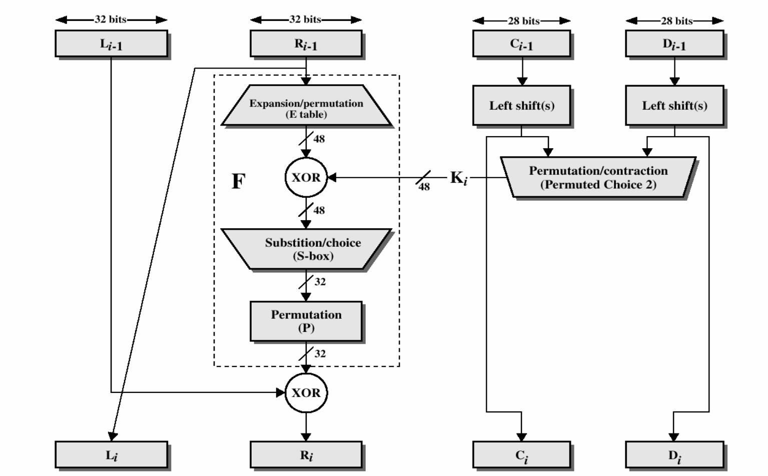


Figure 2.4 Single Round of DES Algorithm

One Round of DES

- Key Transformation
  - Each key-half is shifted 1 or 2 bits in each round (per given table)
  - The 56 key bits are permuted and 48 bits are chosen (per table)
- Text transformations
  - Expansion of R_i from 32 to 48 bits (size of key)
    - Avalanche effect – some bits are duplicated
  - 48 bits are XORed with K_i
  - Substitution, using 8 S-Boxes with 6-bit input and 4-bit output
    - S-boxes are well chosen to introduce non-linearity

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

  - 32 bits are permuted according to specified P-Box
  - 32 bits are XORed with L_i to create R_{i+1}

Data Encryption Standard (DES)

- Software implementations are slow
  - On IBM Mainframe 32,000 blocks / second
- Hardware implementations are very fast
  - VLSI Technology 6868 (“Gatekeeper”) DESes in 8 clock cycles
  - DEC built GaAs gate array that DESes 16.8 million blocks / second
- Weak keys
  - All 0’s, or all 1’s in each half would result in same subkeys
  - Note: if K’=complement of K, then $E_{k'}(P')$ =complement of $E_k(P)$
- There were also claims that the S-boxes were weakened by the NSA
- Notable DES Attacks
  - In 1990, Eli Biham and Adi Shamir presented *differential cryptanalysis*
    - A chosen-plaintext attack that uses two plaintexts with specific difference. Then, based on the difference in the ciphertext (and also internal rounds), one can update the a priori probability of keys
  - In 1993, Mitsuru Matsui showed *linear cryptanalysis* attack
    - Certain XORs of plaintext and ciphertext bits will result in a certain XOR of key bits with some probability $p \neq 1/2$

RC5

- Invented by Ron Rivest (Ron’s Code 5), and developed by RSA Technology into a number of their products
- A block cipher that uses only XORs, Additions, and Rotations
- Variable length blocks, keys, and number of rounds
- A,B are two halves of text; Si are key-based
  - $A=((A \oplus B) <<< B) + S_{2i}$
  - $B=((A \oplus B) <<< A) + S_{2i+1}$
- With 16+ rounds, it resists differential attack
- Uses low-cycles operations, and is very fast

Other Block Ciphers

- Blowfish (Schneier)
  - Simple: additions, XORs, and table lookups
  - Table lookups may require large memory
  - Variable key length
- CAST
  - The round function differs from one round to next
- Int’l Data Encryption Alg (IDEA), Lai and Masey
  - Plaintext, key, and ciphertext are divided to 4 parts
  - Uses XORs, additions, and multiplications in 8 rounds
  - 128-bit key, 52 16-bit subkeys (can be independent)
  - Resists differential cryptanalysis
  - Used in PGP

Triple DES (3DES)

- 3DES uses three 56-bit keys
  - $C=E_{k1}(D_{k2}(E_{k3}(P)))$
  - $P=D_{k1}(E_{k2}(D_{k3}(P)))$
- Note: if K1=K2 then 3DES=DES
- Double encryption doesn’t work well
  - Merkle-Hellman chosen plaintext men-in-the-middle attack requires only 2^{n+1} trials

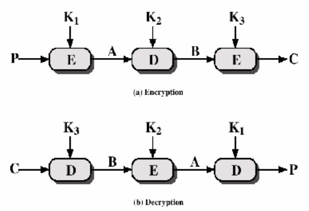


Figure 2.6 Triple DEA

Advanced Encryption Standard (AES)

- NIST put out the RFP in 1997
- Five finalists:

	MARS	RC6	Rijndael	Serpent	Twofish
General Security	3	2	2	3	3
Implementation of Security	1	1	3	3	2
Software Performance	2	2	3	1	1
Smart Card Performance	1	1	3	3	2
Hardware Performance	1	2	3	3	2
Design Features	2	1	2	1	3

- In October 2000, NIST recommended Rijndael

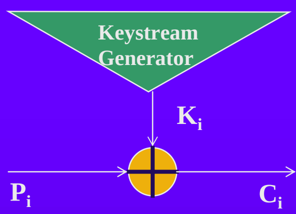
Rijndael Block Cipher

- By Belgians Joan Daemen, and Vincent Rijmen
- Basic operations use bit-coefficient polynomials, in $GF(2^8)$
- Does not use Feistel structure
- Instead uses 3 types of layers and a state
  - Non-linear layer, using optimized S-boxes
  - Linear mixing layer for diffusion of all bits throughout the rounds
  - Key addition layer, using a simple XOR
- Each round
  - Byte substitution (S-box from state matrix, with index (i,j) based on previous state)
  - Row shift (to the matrix of states)
  - Column mix (also to the matrix of states)
  - Key XOR with the current state

Cipher Block Modes of Operation

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Stream Ciphers



- Stream ciphers work on a stream of blocks (sometimes bits), altering the encryption from one block to the next
  - Keystream may change according to original key, previous encryptions, and block index
    - In synchronous stream ciphers, keystream does not depend on text
  - Other encryption parameters may also change, e.g., S-boxes
- E.g. RC4
  - Uses 8x8 S-box, with all possible 8-bit key-entries
  - Keys are selected randomly, and XORed with plaintext to produce ciphertext
  - In each iteration, key entries are switched
  - RC4 is used in Lotus Notes, CDPD, and SSL

Cipher Block Modes of Operation

- Stream ciphers can be implemented from block cipher building blocks
- Requirements:
  - Should be efficient, without significant overhead
  - Shouldn't allow chosen plaintext attacks to interfere with the encryption
  - Should be fault tolerant, not crashing in case of bit errors
- Note that the secrecy depends on the underlying cipher block algorithm

Electronic Codebook (ECB) Mode

- Simplest form
  - Each block (typically 64 bits) encrypted separately
  - As if there is a codebook of 2^{64} entries (per key)
- Fast, easy to parallelize
- Relatively fault tolerant
- Easy target to known-plaintext attack
  - cryptanalyst can rebuild the code book
  - Also susceptible to stereotypical beginning and ending of messages and statistical attacks
- Also easy target to modification attack
  - E.g., replacing the target-account block in a bank money wiring communication

Cipher Block Chaining (CBC) Mode

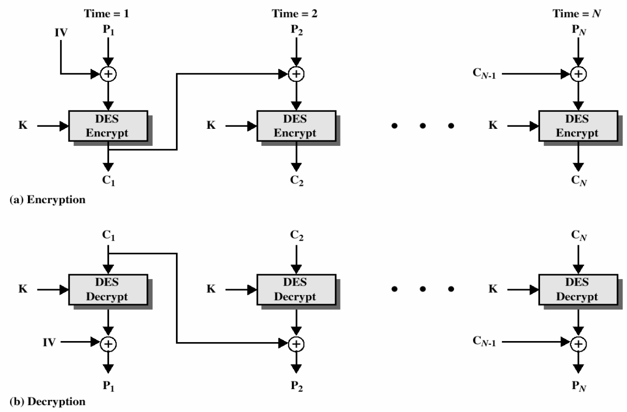
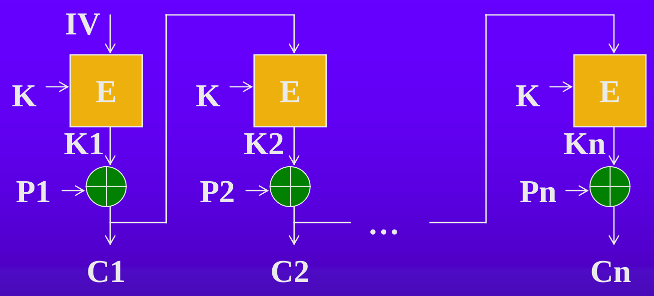


Figure 2.7 Cipher Block Chaining (CBC) Mode

Cipher Block Chaining (CBC) Mode

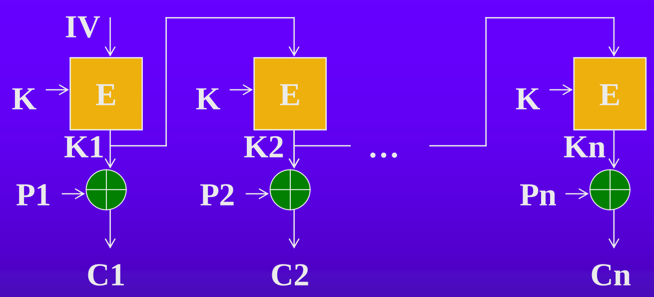
- Encryption
  - $C_i=E_k(P_i \oplus C_{i-1})$
  - $C_0=IV$
- Decryption
  - $P_i=D_k(C_i) \oplus C_{i-1}$
- Initialization vector modifies encryption of identical blocks
  - Can be chosen by source and sent in the clear
  - Or, encrypt random data in the first block
- Errors
  - A bit of error in the plaintext will not extend the error
  - A bit of error in the ciphertext will garble that block, and will alter same bit in the next block, but then CBC self-recovers completely
- Security
  - A man-in-the-middle can easily append blocks in the end
  - Can change a bit, knowing which bit will be affected in 2^{nd} block

Cipher Feedback Mode (CFB)



- Errors
  - A bit of error in plaintext affects all subsequent blocks but does not extend the error when decrypted
  - A bit of error in ciphertext affects same bit and next block, after which CFB self synchronizes

Output Feedback Mode (OFB)



- Output of Encryption serves as feedback

Key Management for Conventional Cryptography

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Key Generation, Distribution and Management

- The security of any cryptographic system depends on safe and effective key distribution and management
  - frequent changes
  - low computational and communication overhead
- Key Distribution Centers (KDCs) are the single most critical point of failure, and are the toughest to implement

- Key Generation
  - Cryptanalyst may attack the key generation algorithm
- Distribution
  - Opponent may impersonate or attack the communication
- Management
  - Adversary may attack KDC systems, or simply exploit human weaknesses

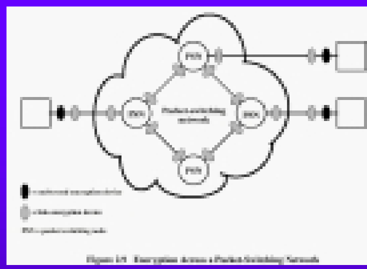
Key Generation

- Key space should be large enough
- Selection from key space shall be random
  - Humans select poor keys prone to dictionary attack
  - Some algorithms have weak keys that should be avoided (DES has 16 such weak keys)
- ANSI X9.17 Key Generation Algorithm
  - Key is generated from previous key, through some encryption process that also takes into account a kept state information
  - Seeds generated from low-order bits of time stamps, time between keystrokes of administrator, etc.

Key Distribution Alternatives

- Physical Delivery
  - Alice can select the key and deliver to Bob
  - Charles, a trusted third-party, can select the key and deliver to both Alice and Bob
- Encrypted direct communication
  - From Alice to Bob using an earlier encrypted session
- Encrypted communication with trusted third-party
  - From Charles to both Alice and Bob

Key Distribution (cont.)



- Encryption location
  - Link encryption
  - End-to-end encryption
- Physical delivery
  - best for link encryption, e.g., routers that link two sub-networks
  - hard for end-to-end, esp. ad-hoc / many-to-one communication
- Encrypted direct key-delivery communication
  - Dangerous: an attacked that gets one key, gets them all
- Conclusion:
  - Security of link communication should not be compromised, and shall use manual delivery of keying material (especially key-encryption keys)
  - End-to-end communication can use key-delivery by third party (data keys)

Session Key Distribution by KDC

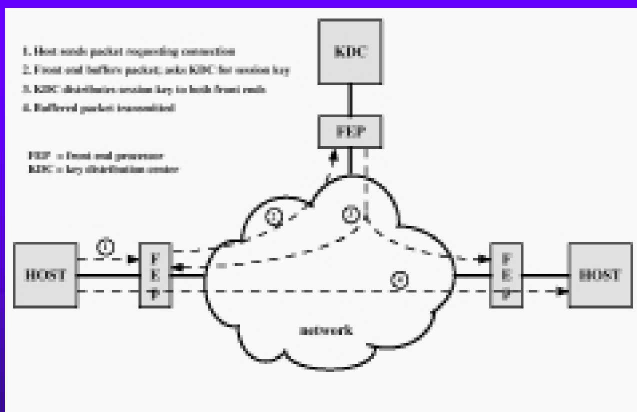


Figure 2.10 Automatic Key Distribution for Connection-Oriented Protocol

- It is safer if KDC-host link encrypted using a physically delivered key
- KDC-host communication shall also be mutually authenticated

Key Management Principles

- To reduce the risk of eavesdropping
  - use different keys for different purposes
  - generate new keys from old ones + hash function
- To reduce the risk of impersonation
  - use mutual authentication when exchanging keys
- To reduce the risk of computer/physical break-in
  - store most keys encrypted using master key
  - save master keys in your memory, smart card, flash key, etc.
  - use tamper-proof hardware encryption, much safer than software
  - destroy media on which keys were stored, even if were encrypted
- Replace keys frequently
- Report compromised keys to KDC with timestamp
- Backup keys shall be broken and spread

Message Authentication

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Message Authentication

- Goal: offer protection against active attacks
  - Impersonation
  - Modification of contents
  - Replay
  - Interruption and denial of service
- Requirements
  - Message is authentic - has not been altered
  - Message source is authentic
  - Optional
    - Message arrived in correct sequence
    - Non-repudiation

Message Authentication Approaches

- Conventional encryption
  - After all, only the parties should have access to key
- Message authentication without encryption
  - Authentication tag is attached to message to verify its integrity and the integrity of the source
- Message Authentication Code (MAC)
  - MAC=F(Message,Key)

Message Authentication Code

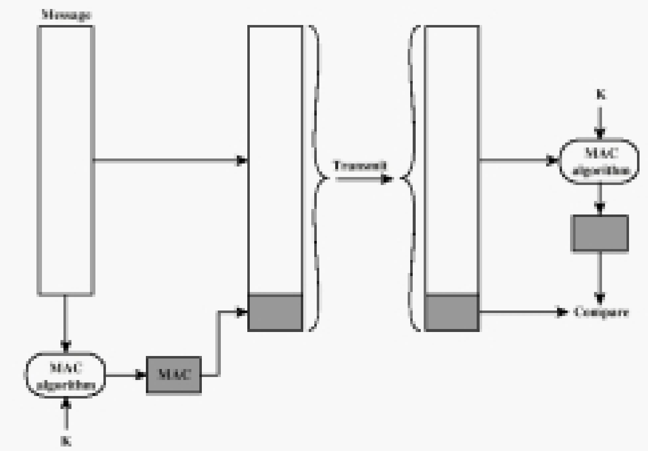


Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

MAC Properties

- Message is authentic
  - If the attacker modified the message, the MAC will likely not match the one calculated by the receiver
- Source is authentic
  - No one else has the key to generate same MAC
  - Hence, also non-repudiation
- Message is in sequence
  - Should add timestamp or other nonce to the message before calculating the MAC

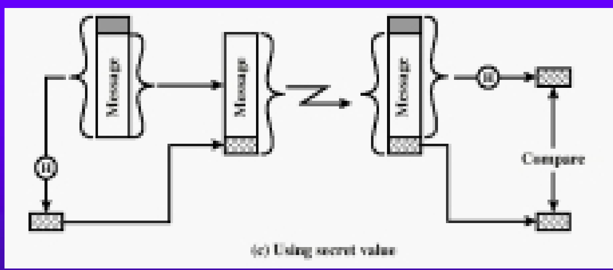
- Any encryption algorithm can be used to generate MAC
  - NIST recommended last *n* bits of DES-encryption of the message

One-Way Hash Functions

- Note that for the purpose of authentication, MAC function need not be reversible
- A *one-way hash function* H, takes an input an arbitrary length message M, and produces a fixed-length hash value
  - H must be easy to compute
  - H is hard to reverse, i.e. given h, its hard to find M
  - H(M) is hard to duplicate , i.e., it is possible that there exists M' such that H(M)=H(M'), but given M it hard to find such M'
- For some applications, we may need *collision resistance*:
  - It is hard to find arbitrary M and M' such that H(M)=H(M')
- H(M) is a fingerprint of the message M and is called *message digest (MD)*

Message Authentication Protocol Using a One-Way Hash Function

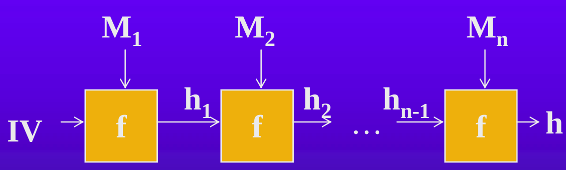
1. Using a symmetric secret / key



2. Using symmetric encryption
   - Generate H(M), which is small in size
   - Use $E_K(H(M))$ as the MAC

Construction of One-Way Hash Functions

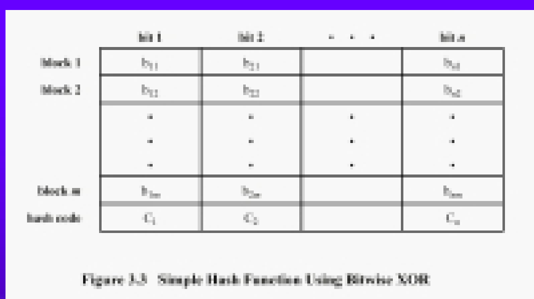
- Hash functions are typically based on compression functions (f) that work on blocks (Mi)



- Works like a chained block cipher
  - Produces a hash value for each fixed-size block based on its content and based on the hash value for the previous block
- In fact, can use symmetric encryption as f=E, and use M_i as the key

Simple Hash Functions

- Bitwise-XOR



	bit 1	bit 2	. . .	bit n
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block m	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

Figure 3.3 Simple Hash Function Using Bitwise XOR

- Not very secure, e.g., for English text (ASCII<128) the high-order bit is almost always zero
- Can be improved by rotating the hash code after each block is XORed into it
- Still, if the message itself is not encrypted, it is easy to modify the message and append one block that would set the hash code as needed

Secure Hash Algorithm (SHA)

- SHA was published by NIST as a standard in 1993
- Revised in 1995 as SHA-1
  - Input: Up to 2^{64} bits
  - Output: 160 bit digest
- Pad with at least 64 bits to resist padding attack
  - 1000…0<message length>
- Processes 512-bit block
  - Initiate 5x32bit MD registers
  - Apply compression function
    - 4 rounds of 20 steps each
    - each round uses different non-linear fi
    - registers are shifted and switched

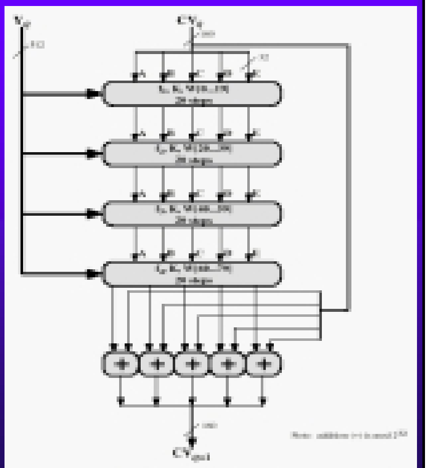
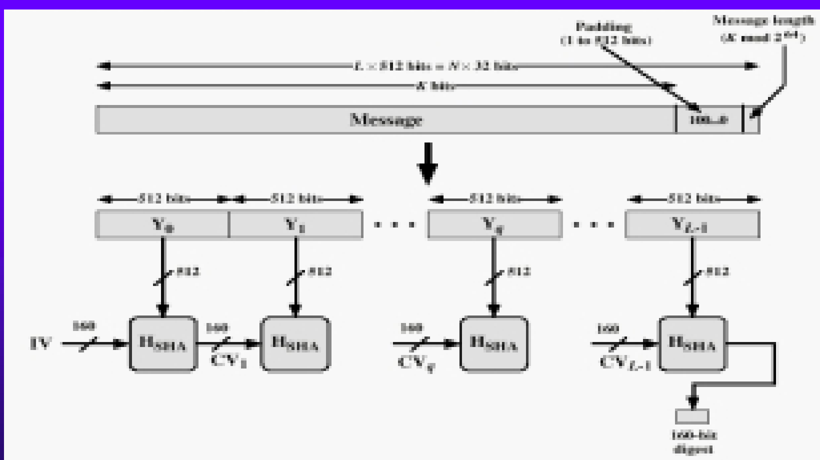


Figure 3.5 SHA-1 Processing of a Single 512-bit Block

SHA-1



Other Famous MD Algorithms

	SHA-1	MD5 (MD4+	RIPEMD-160
Digest length	160 bits	128 bits	160 bits
Basic unit of processing	512 bits	512 bits	512 bits
Number of steps	80 (4 rounds of 20)	64 (4 rounds of 16)	160 (5 paired rounds of 16)
Maximum message size	2^{64}-1 bits	unlimited	unlimited

Variable Length Hash Codes

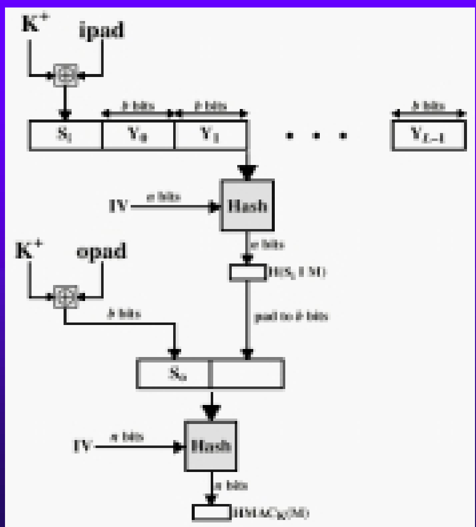
- Some hash functions have good cryptographic qualities, but generate short hash codes
  - If the message digest is short, the receiver can easily forge another message with same hash code
  - Similarly, easy to find a (message,hashcode) pair that match
- Can use the following algorithm to enlarge hash code
  - Start with M0=M, H0=H(M)
  - Generate M1 by appending H0 to M0, and generate H1=H(M1)
  - Append H1 to H0
  - Repeat until generated enough hash codes

Hash Function MAC (HMAC)

- **HMAC Idea**: Use a MAC derived from any cryptographic hash function
  - Note that hash functions do not use a key, and therefore cannot serve directly as a MAC
- **Motivations for HMAC:**
  - Cryptographic hash functions execute faster in software than encryption algorithms such as DES
  - No need for the reverseability of encryption
  - No export restrictions from the US
- **Status**: designated as mandatory for IP security
  - Also used in Transport Layer Security (TLS), which will replace SSL, and in SET

HMAC Algorithm

- Compute H1= H of the concatenation of M and K1
- To prevent an “additional block” attack, compute again H2= H of the concatenation of H1 and K2
- K1 and K2 each use half the bits of K
- Notation:
  - K^+ =K padded with 0’s
  - ipad=00110110 x b/8
  - opad=01011100 x b/8
- Execution:
  - Same as H(M), plus 2 blocks



Public-Key Cryptography

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Motivation

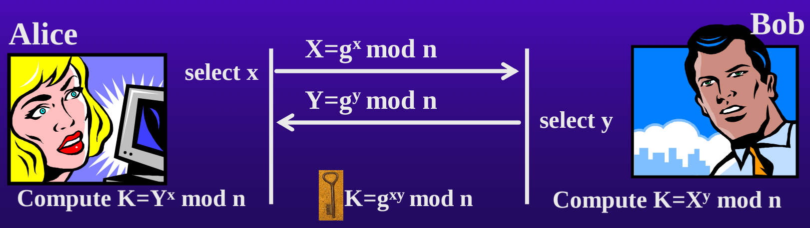
- Until early 70s, cryptography was mostly owned by government and military
- Symmetric cryptography not ideal for commercialization
  - Enormous key distribution problem; most parties may have never physically met
  - Must ensure authentication, to avoid impersonation, fabrication
- Few researchers (Diffie, Hellman, Merkle), in addition to the IBM group, started exploring Cryptography because they realized it is critical to the forthcoming digital world
  - Privacy
  - Effective commercial relations
  - Payment
  - Voting

Public-Key Cryptography

- First proposed by Diffie and Helllan, and independently by Merkle (1976)
  - Idea: use separate keys to encrypt and decrypt
  - Merkle proposed puzzles, and then knapsack problems
- Pair of keys is generated by each user
  - Public key is advertised
  - Private key is kept secret, and is computationally infeasible to discover from the public key and ciphertexts
  - Each key can decrypt messages encrypted using the other key
- Applications:
  - Encryption
  - Authentication (Digital Signature)
  - Key Exchange (to establish Session Key)

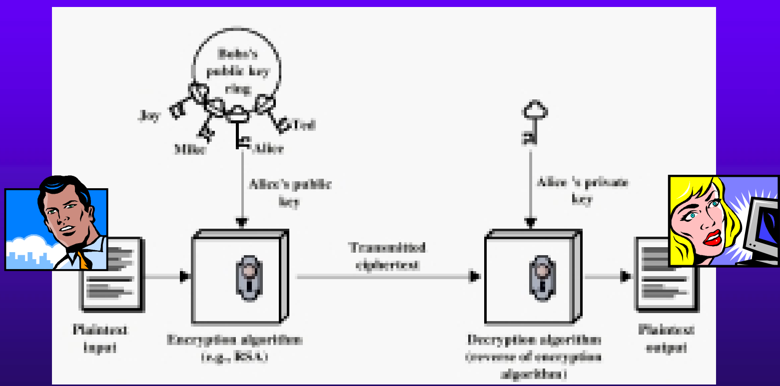
Diffie-Hellman Key Exchange

- First public-key algorithm, based on the difficulty of computing discrete logarithms modulo n
- Protocol:
  - Use key exchange protocol to establish session key
  - Use session key to encrypt actual communication
- Algorithm:
  - Choose a large prime n, and a primitive root g



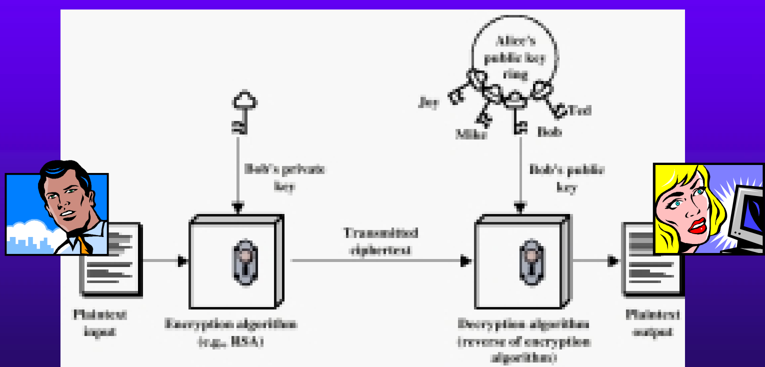
Public-Key Encryption

- Sender uses the public key of the receiver to encrypt
- Receiver uses her private key to decrypt



Authentication Using Public-Key

- The sender encrypts the message with his own private key
- The receiver, by decrypting, verifies key possession



Public-Key Algorithms: Requirements

- It is computationally easy to generate a pair of keys
- It is computationally easy to encrypt using the public key
- It is computationally easy to decrypt using the private key
- It is computationally infeasible to compute the private key from the public key
- It is computationally infeasible to recover the plaintext from the public key and ciphertext
- Either of the related keys can decrypt a message encrypted using the other key

- Note: it should be computationally infeasible to decrypt using same key used for encryption

RSA

- Developed by Rivest, Shamir, and Adleman (1977), and is most widely used
  - Classified version of RSA developed by GCHQ (Ellis and Cocks) in 1973

- Gets its security from the difficulty of factoring large numbers
- Works as a block cipher, where each plaintext/ciphertext block is integer between 0 and *n*
- Algorithm:
  - Receiver chooses *e*, *d*
  - The values of *e*, and *n* are made public; *d* is kept secret
  - Encryption: $C=M^e \bmod n$
  - Decryption: $M=C^d \bmod n = M^{ed} \bmod n$
- Requisite:
  - Find *e*, *d* such that $M=M^{ed} \bmod n$, for all $M<n$
  - Make sure that *d* cannot be computed from *n* and *e*, not even if a ciphertext is available

RSA Key Generation

- Select primes p and q, $n=pq$
- Calculate $\Phi(n)=(p-1)(q-1)$
  - Euler totient of n – number of integers between 1 and n that are relatively prime to n, i.e., $\{m \mid gcd(m,n)=1\}$
- Select integer $e<\Phi(n)$ such that $gcd(\Phi(n),e)=1$
- Calculate d such that $d=e^{-1} \bmod \Phi(n)$,
  - i.e. $ed=1 \bmod \Phi(n)$
- Note:
  - The message could have been encrypted with d and decrypted by e

RSA Key Generation: Why it Works

- Fermat's Little Theorem
  - For a prime p, $\forall a$ such that $0<a<p$, $a^{(p-1)}=1 \bmod p$
- Euler's extension
  - For primes p,q, $\forall a$ such that $gcd(a,pq)=1$, $a^{(p-1)(q-1)}=1 \bmod pq$
  - Hence, $M^{ed} \bmod n = M^{k(p-1)(q-1)+1} \bmod n = 1xM = M$
- To generate primes, use primality test
  - For a non-prime, Fermat's theorem will usually fail on a random a
    - Carmichael numbers are very rare exception, and if chosen decryption wont work. Can reduce the probability by checking more a's
  - Primes are dense enough (almost one of every k k-bit numbers)
- GCD to select e takes $O(\log n)$ time
- Calculate $d=e^{-1} \bmod n$ using Euler extended GCD algorithm
- Exponentiation (Encrypt/Decrypt) takes $O(\log n)$ time
- RSA gets its security from the difficulty of factoring $n=pq$

RSA Example

- Key Generation
  - Select p=7, q=17, n=pq=119, Φ(119)=96
  - Select e=5; Calculate d=77

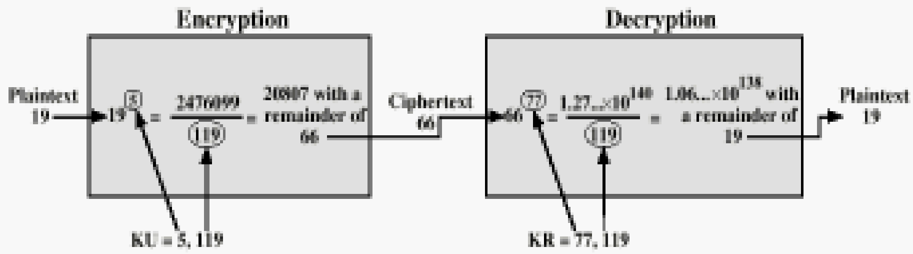


Figure 3.9 Example of RSA Algorithm

Attacks on RSA Algorithm

- If one could factor n, which it available, into p and q, then d could be deduced, and then the message deciphered
- If one could guess the value of $(p-1)(q-1)$, even without factoring n, then again d could be deduced

Attacks on RSA Protocol

- Chosen ciphertext attack
  - Attack: get sender to sign (decrypt) a chosen message
  - Inputs: original ciphertext $C=M^e$
  - Construct
    - $X=R^e \bmod n$, for a random R
    - $Y=XC \bmod n$
    - $T=R^{-1} \bmod n$
  - Ask sender to sign Y, obtaining $U=Y^d \bmod n$
  - Compute
    - $TU \bmod n = R^{-1}Y^d \bmod n = R^{-1}\, X^d\, C^d \bmod n = C^d \bmod n = M$
  - Exploits preservation of multiplication under mod
- Conclusion:
  - never sign a random message
  - sign only hashes
  - use different keys for encryption and signature

Other precautions when implementing RSA protocol

- Do not use same *n* for multiple users
  - Can decipher using two encryption (public) keys, without any decryption key
- Always pad messages with random numbers, making sure that M is about same size as *n*
  - If e is small, there is an attack that uses *e(e+1)/2* linearly dependent messages
- Do not choose low values for *e* and *d*
  - For e, see above, and there is also attack on small *d*'s

Other Public-Key Algorithms

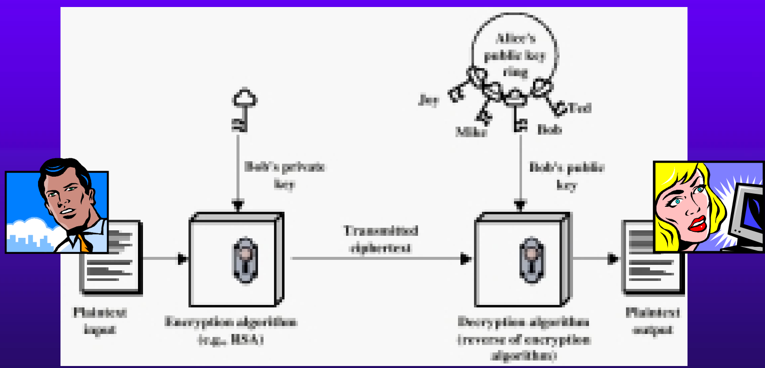
- Merkle-Hellman Knapsack Algorithms
  - First public-key cryptography algorithm (1976)
  - Encode a message as as series of solutions to knapsack problems (NP-Hard). Easy (superincreasing) knapsack serves as private key, and a hard knapsack as a public key.
  - Broken by Shamir and Zippel in 1980, showing a reconstruction of superincreasing knapsacks from the normal knapsacks
- Rabin
  - Based on difficulty of finding square roots modulo n
  - Encryption is faster: $C=M^2 \bmod n$
  - Decryption is a bit complicated and the plaintext has to be selected from 4 possibilities
- El Gamal
  - Based on difficulty of calculating discrete logarithms in a finite field
- Elliptic Curves can be used to implement El Gamal and Diffie-Hellman faster

Digital Signatures

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Public-Key Digital Signature

- The sender encrypts the message with his own private key
- The receiver, by decrypting, verifies key possession



Digital Signatures

- The entire message, encrypted with the private key, serves as the digital signature
  - Computationally expensive
  - Anyone can decrypt the original message
- Alternatively, a *digest* can be used
  - Should be short
  - Prevent decryption of the original message
  - Prevent modification of original message
  - Difficult to fake signature for
- A hash code of the message (e.g., SHA-1)
- If only source authentication is needed, a different message can be used

Digital Signature Algorithm (DSA)

- Proposed in 1991 by NIST as a standard (DSS)
- Based on difficulty of computing discrete logarithms (like Diffie-Hellman and El Gamal)
- Encountered resistance because RSA was already de-facto standard
  - Cannot be used for encryption or key distribution
  - Faster than RSA in signature, but slower in verification
  - Significant investment in RSA by large corporations
  - Concerns about NSA backdoor
- Key size was increased from 512 to up-to 1024 bits

Description of DSA

- Public parameters
  - p is a prime number with up to 1024 bits
  - q is a 160-bit factor of (p-1), and itself prime
  - $g=h^{(p-1)/q} \bmod p$
  - x is the private key and is smaller than q
  - $y=g^x \bmod p$ is the public key
  - H(M) is the secure hash code of the message
- Signature
  - Generate a random $k<q$
  - Compute and send $r=(g^k \bmod p) \bmod q$
  - Compute and send $s=k^{-1}(H(M)+xr) \bmod q$
- Verification
  - Compute $w=s^{-1} \bmod q$
  - Compute $u1=H(M)w \bmod q$; $u2=rw \bmod q$
  - Compute $v=(g^{u1}*y^{u2} \bmod p) \bmod q$
  - If v=r then the signature is verified

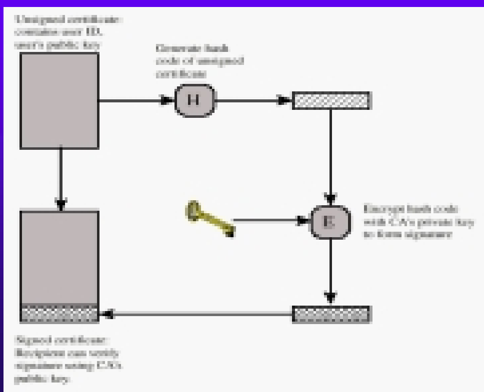
Key Management for Public-Key Cryptography

Main sources: Network Security Essential / Stallings
Applied Cryptography / Schneier

Certificate Authority: Verifying the Public Key

- How to ensure that Charles doesn't pretend to be Bob by publishing a public-key for Bob. Then, using a Man-in-the-Middle attack, Charles can read the message and reencrypt-resend to Bob

- Bob prepares certificate with his identifying information and his public key (X.509)
- The Certificate Authority (CA) verifies the details and sign Bob's certificate
- Bob can publish the signed certificate



More on Key Management

- Alice may have more than one key
  - e.g., personal key and work key
- Where shall Alice store her keys
  - Alice may not want to trust her work administrator with her personal banking key
- Distributed certification V1.0
  - CA certifies Agents who certify companies who certify employees
- Distributed Certification V2.0 (a la PGP)
  - Alice will present her certificate with “introducers” who will vow for her
- Key Escrow
  - US American Escrowed Encryption Standard suggests that private keys be broken in half and kept by two Government agencies
  - Clipper – for cellular phone encryption
  - Capstone – for computer communication